INTERWOVEN

# TeamSite®
# Backing Store Conversion Planning Guide

## Release 5.5.1

**for Windows NT® and Windows® 2000**

INTERWOVEN

Interwoven, Inc.
803 11th Ave.
Sunnyvale, CA 94089

http://www.interwoven.com

Printed in the United States of America
Part # 10-00-10-11-00-551-010

# Table of Contents

Chapter 1

# Introduction

This document is designed to help you plan the migration of your current old-format backing store (created with pre-5.5.1 version of TeamSite®) to the new high-performance backing store format required by TeamSite 5.5.1.

**Note:** Your old backing store can be created with any pre-5.5.1 version of TeamSite, but you must be running version 4.5.x or 5.0.x to perform the conversion.

This document is distributed independent of TeamSite 5.5.1 to customers using earlier versions of TeamSite. It contains an overview of the new backing store format, the requirements for backing store conversion, several approaches to help you decide what and how to convert, and a section on organizing and managing your new backing store.

Before beginning your conversion, check `http://support.interwoven.com` to ensure that you have the latest conversion tools and the most recent version of this document.

This chapter is organized as follows:

- Backing Store Overview
- High-Performance Backing Store Format
- MultiStore Support
- Backing Store Conversion Tools
- Backing Store Administration Tools
- Documentation Conventions

## Backing Store Overview

The backing store is a large directory structure created by the TeamSite installation program that contains TeamSite files and metadata. By default, the backing store is located in `C:\iw-store`.

Previous releases of TeamSite have been limited to one backing store per TeamSite server. This release supports as many as eight backing stores per TeamSite server (the first of which is created automatically by the installation program, the others by your TeamSite system administrator). These backing stores can be located on different file systems, local to the TeamSite server machine. The functionality that enables multiple backing stores is known as *MultiStore*.

To include MultiStore support in TeamSite (and improve overall server performance), a new backing store format needed to be implemented. This format is used by all backing stores created using the current TeamSite release. If you have a backing store created with TeamSite version 4.5.x or 5.0.x, you must convert the old backing store to use the new format.

Dividing your existing backing store into new multiple stores (possibly on different file systems) enables you to simplify data management, including faster data backup. It also avoids having your backing stores grow to unmanageable sizes.

**Note:** You can migrate data to your new stores any way you choose, but the data between the stores is completely independent and may not be migrated to other stores using inter-branch copying. Copies remain branch-specific and cannot be used at the backing store level.

Backing stores have a corresponding archive in the `VPATH`. In previous versions of TeamSite, there was only one archive named `default` with a corresponding backing store called `iw-store\default`. MultiStore functionality allows for multiple backing stores with user-assigned names. Each backing store is similar to the `default` archive in that it contains a single root branch called `main` and is independent of any other store controlled by the server. All mounted backing stores are assigned a unique store ID number and maintain their own file system resources that are stored persistently inside each backing store.

Backing stores which are named using multibyte characters must be created by editing the `iw.cfg` file. For more information, see “Defining Backing Stores in the iw.cfg File” on page 32.

## High-Performance Backing Store Format

TeamSite backing stores have been redesigned to contain fewer small metadata files, and to use less disk space and fewer file system resources which results in better overall performance. All new backing stores created with this version of TeamSite will use the new format. Old-format backing stores must be converted to the new format before they can be used. Backing store conversion tools are included to convert old backing stores to use the new high-performance format.

## MultiStore Support

TeamSite now includes support for as many as eight active backing stores with each store mapping to one physical file system location. Each backing store contains its own set of branches, workareas, and editions. Multiple stores associated with a TeamSite server share the same set of TeamSite configuration files (for example, users and roles).

TeamSite users using the file system interface, WebDesk, or WebDesk Pro will be able to see all the active stores (to which they have permission to access) associated with the TeamSite server against which they are running.

## Backing Store Conversion Tools

The TeamSite installation program installs a backing store conversion program (`iwconvert`) that you can run from the command line or from the included GUI. The program converts old-format backing stores (those created with TeamSite version 4.5.x or 5.0.x) to the new high-performance format.

You can also use `iwconvert` to divide your single old-format backing store into multiple new-format backing stores (for example, dividing it up by branch or edition, or separating the stores for an intranet site and an Internet site). For detailed information about the `iwconvert` CLT, refer to page 26.

Another CLT, `iwmigrate`, is included to convert a single new-format backing store into multiple new-format stores. For detailed information about the `iwmigrate` CLT, refer to page 38.

## Backing Store Administration Tools

In addition to the backing store conversion tools, the TeamSite installation program installs a number of command-line tools (CLTs) for managing and creating backing stores. Administrative functions including creating, deleting, freezing, activating, and deactivating backing stores are performed by the `iwstoreadm` CLT. For detailed information about the `iwstoreadm` CLT, refer to page 36.

## Documentation Conventions

This manual uses the following notation conventions:

| Convention | Definition and Usage |
|---|---|
| **Bold** | Text that appears in a GUI element (for example, a menu item, button, or element of a dialog box) and command names are shown in bold. For example:<br>Click **Edit File** in the Button Bar. |
| *Italic* | Book titles appear in italics.<br>Terms are italicized the first time they are introduced.<br>Important information may be italicized for emphasis. |
| `Monospaced` | Commands, command-line output, and file names are in monospaced type. For example:<br>The `iwextattr` command-line tool allows you to set and look up extended attributes on a file. |

| Convention | Definition and Usage |
|---|---|
| *`Monospaced italic`* | Monospaced italics are used for command-line variables. The most common example of this is *`iw-home`*, which refers to the directory where TeamSite is installed. For example:<br>*`iw-home`*`\etc\iw.cfg`<br>is the path to the main TeamSite configuration file, `iw.cfg`, which is located in the `etc` directory under the TeamSite installation directory.<br>`iwckrole` *`role user`*<br>means that you must insert the values of *`role`* and *`user`* yourself. |
| **`Monospaced bold`** | Monospaced bold represents user input. The `>` character that appears before a line of user input represents the command prompt and should not be typed. For example:<br>`>`**`iwextattr -s project=proj1 //IWSERVER/default/main/dev/WORKAREA/andre/products/index.html`** |
| ***`Monospaced bold italic`*** | Monospaced bold italic text is used to indicate a variable in user input. For example:<br>`>`**`iwextattr -s project=`*****`projectname workareavpath`***<br>means that you must insert the values of *`projectname`* and *`workareavpath`* when you enter this command. |
| `[]` | Square brackets surrounding a command-line argument mean that the argument is optional. |
| `\|` | Vertical bars separating command-line arguments mean that only one of the arguments can be used. |

Chapter 2

# Planning the Conversion

This chapter describes the planning process that should precede the conversion of your existing TeamSite backing store. The information is organized as follows:

- Conversion Overview
- Conversion Prerequisites and Tips
- Conversion Strategies
  - What Do I Convert?
  - Which Conversion Tool Do I Use?
  - Which Conversion Methodology Do I Use?
  - How Do I Organize My New Backing Store?
- SID Changes to the TeamSite Backing Store

# Conversion Overview

This section describes the conversion procedure in very general terms. It is intended to help you understand what is involved in the conversion procedure before you begin. The detailed procedure is included for reference in Chapter 3, "Conversion Procedures and Tools."

1. Satisfy the prerequisites (as described on page 13).
2. Decide how you want to organize your new backing store on the target system:
   - Convert your single old-format backing store into a single new-format backing store.

     You can later convert your single new-format backing store into multiple new-format backing stores by using the `iwmigrate` CLT as described on page 38.
   - Convert your single old-format backing store into multiple new-format backing stores.
3. Run `iwfsck -d` to diagnose and repair any potential problems in your current backing store. The `iwfsck` CLT is documented in the *TeamSite Command-Line Tools* manual.
4. Run the `iwconvert` conversion tool from either the GUI or the command line.

   You may repeat this step any number of times depending on what you plan to convert from your existing store, and whether you plan to create multiple new-format stores.

**Optional Steps:**

5. Make an edition of your staging area after the initial conversion pass (or passes) is complete.

   This edition will contain anything submitted while the initial conversion was running. It should be much smaller than your other editions and should convert faster.
6. Run the `iwfreeze` CLT to prohibit any more changes to the staging area.
7. Create and convert the final edition.
8. If you want to use the source system (where your current TeamSite 4.5.x or 5.0.x installation resides) as your production server *after* the conversion is complete, you will need to install TeamSite 5.5.1 on this server and copy the converted (new-format) backing stores onto this server.

# Conversion Prerequisites and Tips

- You must have two systems running the same operating system (they do not need to be the same version) and be located on the same network.

  This document refers to these as the *source* system (which contains your current, old-format backing store) and the *target* system (which will host the new-format version of the backing store.
- Ensure that TeamSite 4.5.x, or 5.0.x is installed and licensed on the source system.
- Back up your existing installation and backing store.
- Ensure that TeamSite 5.5.1 is installed and licensed on the target system.
- The target system must have at least as much disk space as the source system.
- Decide how you are going to divide the source backing store for conversion. Depending on the size of your source backing store, and the organization of your TeamSite implementation, you may choose to convert a range of editions, a single edition, or branch-by-branch.
- Before running the `iwconvert` command, run `iwfsck -d` on your source backing store to prepare for conversion. The `iwfsck` CLT is described in the *TeamSite Command-Line Tools* manual.
- Allow plenty of time for the conversion to complete. While there are many variables affecting the time it takes to convert a backing store, tests show that the conversion runs at about 500 megabytes (MB) per hour. Note that this is a very general number and you should not be concerned if your conversion runs at a different rate.
- The user who mounts `iw-store` and `iwserver` on the source machine, must be a domain user who is in the Administrators group on both the source and target machines.
- Do not interrupt a conversion once it has been started because the target backing store will be corrupted (including previously converted portions of the conversion located in the same backing store).
- Plan to publish new editions of the staging area if you allow users to submit files while the conversion is running. You may have to do this multiple times. Eventually, you will need to freeze your source server to prohibit users from using TeamSite during the final conversion.
- You can use the `iwmigrate` CLT to migrate data between new backing stores. You can also use your operating system's copy functionality, but you will lose history and version information.

- You can have a maximum of eight active backing stores on your target system. You can have more than eight backing stores, but only eight can be active.
- You can create backing stores on NFS-mounted remote servers if you are using disk management devices.
- To avoid having the backing store conversion procedure create an unknown user on the target system, the user doing the conversion must have a user account with access to every file and ACL on both systems.

  If an unknown user is created, you can use the `iwidmap` CLT (as described on page 38) to remap the unknown user to an appropriate user on the target system.
- Workflow tasks must be completed prior to conversion.

## Conversion Strategies

You have a number of options to consider while planning your backing store conversion. These considerations can be summarized as:

- What Do I Convert?
- Which Conversion Tool Do I Use?
- Which Conversion Methodology Do I Use?
- How Do I Organize My New Backing Store?

These questions are examined in the sections that follow.

### What Do I Convert?

Careful planning of what gets converted (and when) is important because users can still edit and submit files in TeamSite while the conversion is running. Depending on the size of your source backing store, and the organization of your TeamSite implementation (single or multiple new-format backing stores), you may choose to convert a single edition, a range of editions, or branch-by-branch.

**Note:** The conversion tools also enable you to convert workareas, but if you have a large number of workareas, organizing active users so that no data is left unconverted can prove to be difficult.

At a minimum, you will need to convert the latest edition on each branch, and any files that are submitted to the staging area while the initial conversion is running.

You can retain your old backing store as a read-only archive on your source system if you keep the old TeamSite (TeamSite version 4.5.x, or 5.0.x) installation operational.

## Which Conversion Tool Do I Use?

You have the option of performing your backing store conversion either from the command line or by using the browser-based conversion GUI. Regardless of which interface you select, the `iwconvert` CLT performs the actual conversion.

If you generally prefer working from the command line, chances are you will find the command-line interface suitable for your conversion needs.

The GUI reduces the amount of typing and displays a view of the file system so it may be easier to use to navigate your file system. You can also use the GUI in a mode where it generates a script, and then you use the script to run `iwconvert`. This script can be easily modified when you begin work on another branch, or if for any reason you need to start over.

## Which Conversion Methodology Do I Use?

You can you use the following scenario to convert the most recent editions on each branch in your backing store. You need to create interim editions if you allow users to work in TeamSite while the conversion is in progress. This method avoids having to convert each workarea on a branch.

1. On any branch, publish an edition of the contents of the staging area.
2. Begin the conversion of this edition.

   This edition contains the most recent version of the files contained in your site. If possible, begin this conversion during a period of low usage.
3. Publish an edition of your staging area after the initial conversion pass is complete.

   This edition will contain anything submitted while the initial conversion was running. It should be much smaller than your other edition and should convert faster.

4. Run the `iwfreeze` CLT to prohibit any more changes to the `STAGING` area.

   During this point in the conversion process, TeamSite users will *not* be able to submit files to the staging area.

5. Convert the edition created in step 3.

6. Start the TeamSite server and allow users to submit their changes.

7. Ensure that all files being modified are submitted to the staging area, and run the `iwfreeze` CLT again to prohibit changes to the staging area.

8. Create and convert the final edition.

9. Run the `iwconvert -c` CLT to clean up the new backing store history.

10. Repeat the procedure for all other branches you want to convert.

This methodology will greatly reduce the overall space required for your backing store, but will not retain the full version history of your files.

## How Do I Organize My New Backing Store?

Decide how you want to organize your new backing store on the target system. You have two options:

- A single new-format backing store—Produces a backing store model similar to your current TeamSite implementation.
- Multiple new-format backing stores—MultiStore functionality enables as many as eight active backing stores per TeamSite server.

  You can implement the MultiStore option in either of two ways: by converting your single old-format backing store directly into multiple new-format backing stores, or by dividing your previously converted single new-format backing store into multiple new-format backing stores using the `iwmigrate` CLT.

These conversion options are depicted in the graphic on page 18.

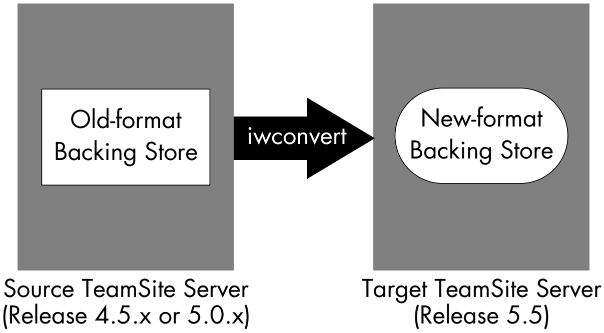
Consider creating separate backing stores for branches that meet the following criteria:

- Distinct deployment targets
- Legacy or infrequently accessed data

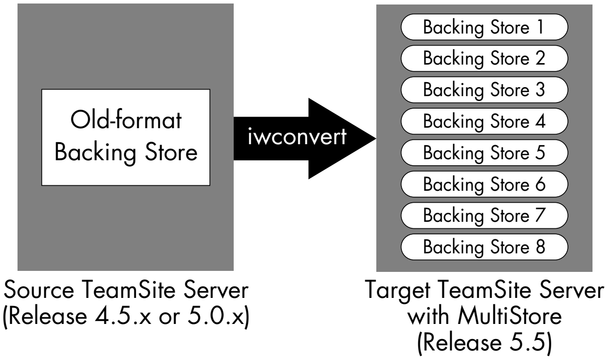
- Distinct ownership within your organization
- Content that will not be shared

**Note:** In addition to creating multiple backing stores by using `iwconvert` or `iwmigrate`, you can add additional new-format backing stores (up to a total of eight) by using the `iwstoreadm` CLT after the conversion process is complete. The `iwstoreadm` CLT is described on page 36.

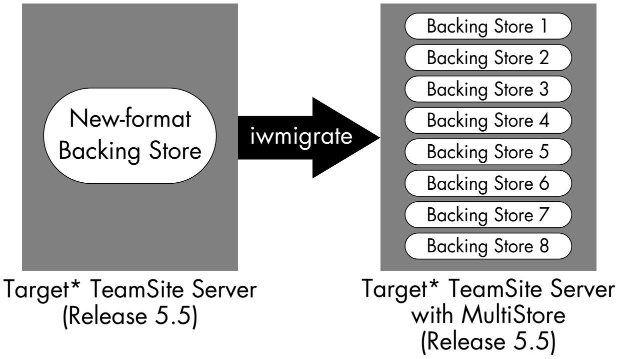
## Old-format Single Store → New-format Single Store


Old-format Backing Store
iwconvert
New-format Backing Store
Source TeamSite Server (Release 4.5.x or 5.0.x)
Target TeamSite Server (Release 5.5)

## Old-format Single Store → New-format Multiple Stores



## New-format Single Store → New-format Multiple Stores



* The iwmigrate procedure can be performed on one TeamSite Server; iwconvert requires two systems.

## Upgrading Your Original Server to TeamSite 5.5.1

If you prefer to use the same server that hosted your TeamSite 4.5.x or TeamSite 5.x installation as your TeamSite 5.5.1 deployment server, you will need to perform an upgrade installation on that server *after* the backing store conversion is complete. The upgrade scenario is as follows:

1. With TeamSite 4.5.x or TeamSite 5.x installed on your current system (call it system A), install TeamSite 5.5.1 on a second system (system B).
2. Convert your old-format backing store (on system A) to the new high-performance backing store format on the system B. (This procedure is described in Chapter 3.)
3. If you want to use System A as your TeamSite 5.5.1 deployment server (as you have been), you must upgrade System A to TeamSite 5.5.1 and also migrate the new backing store (or stores) to system A.

**Note:** If you copy your backing store, ensure that all file attributes—including security and file times—are preserved.

The procedure for this upgrade is described in the Installation chapter of the *TeamSite Administration Guide*.

## SID Changes to the TeamSite Backing Store

Previous releases of TeamSite have stored Windows Security Identifiers (SID) representing users and groups directly in the backing store. This causes problems when converting the backing store onto different systems.

The new-format backing store uses a unique 32-bit ID generated by the TeamSite server instead of storing the SID. A one-to-one persistent mapping exists between the TeamSite generated ID and the SID. Whenever the SID has to be written out to the backing store, the mapping is checked to obtain the TeamSite ID, which is substituted for the SID. When a restore is attempted, a reverse lookup takes place and the appropriate SID is recovered.

A command-line tool called `iwidmap` is included to change the mapping between the SID and the token. It can also be used to refresh the mapping when the same names are used, but the SID has changed. Details about the `iwidmap` CLT are included on page 38.

Chapter 3

# Conversion Procedures and Tools

This chapter describes the actual procedures and CLTs for converting your backing store to the new-format backing store. The information is organized as follows:

- Converting Backing Stores Using the GUI
- Converting Backing Stores from the Command Line
- Creating Multiple Backing Stores
- Administration CLTs

## Converting Backing Stores Using the GUI

Whether you use the conversion GUI or convert from the command line, the backing store conversion is done by the `iwconvert` program. This utility can be run directly from the command line (as described in “Converting Backing Stores from the Command Line” on page 26), from a script generated by the conversion GUI, or interactively by using the conversion GUI. The bulk of the conversion is done while your existing TeamSite server is running, but you will need to freeze your source server (or prohibit users from using TeamSite) for the final edition to ensure that no data is lost.

The backing store conversion GUI is a CGI program (`iwconvert.cgi`) that is installed by the TeamSite installation program and displayed in your Web browser. It is supported by a process called `iwconvertserver` that communicates with remote TeamSite servers and invokes `iwconvert` on behalf of the GUI. You must run the `iwconvertserver` process manually for the GUI to function properly.

Complete the following procedure to convert your old-format backing store to use the new backing store format:

1. Run the `iwconvertserver` utility directly from the command line in `iw-home\bin`, for example:

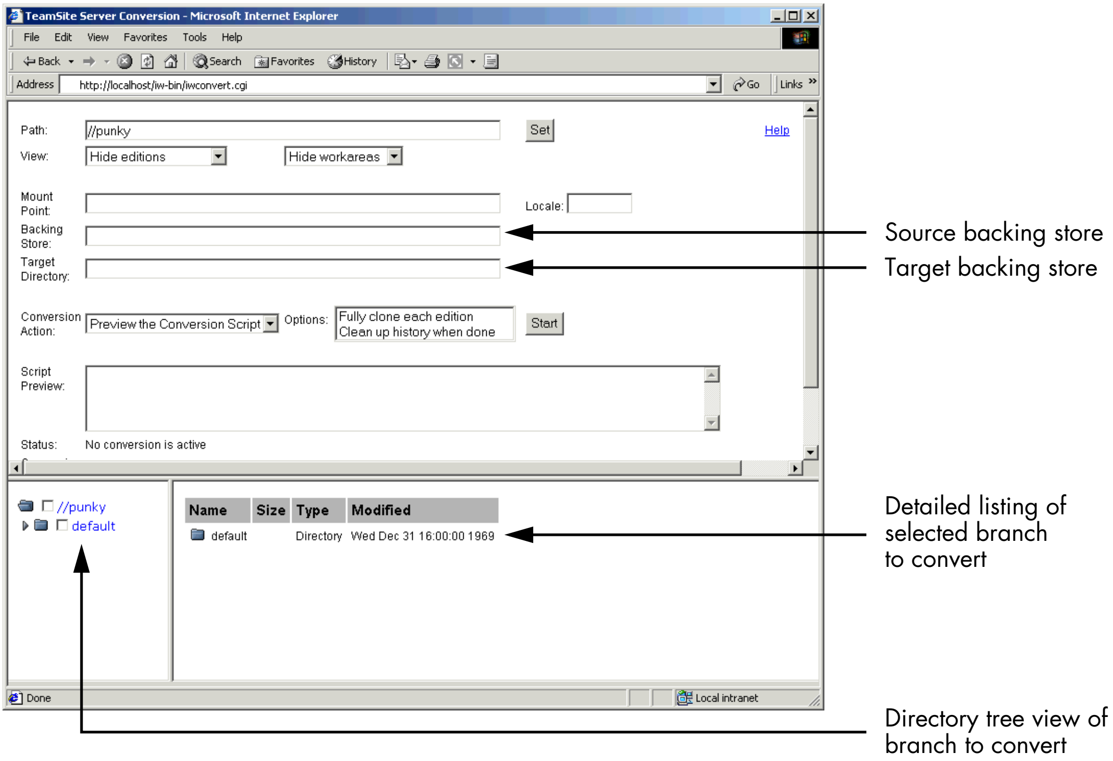
   ```
   iwconvertserver -S
   ```

2. Open your Web browser.

3. Type **`http://localhost/iw-bin/iwconvert.cgi`** in the **Location** field (Netscape) or the **Address** field (Internet Explorer).

   **Note:** If you are not working on the same computer that will contain the new backing store, you must specify the system name instead of `localhost` in this step. For example, if you wanted the new backing store to be located on a server named `factotum`, type: **`http://factotum/iw-bin/iwconvert.cgi`**

The conversion GUI is displayed in your default browser:



TeamSite Server Conversion - Microsoft Internet Explorer

Address: http://localhost/iw-bin/iwconvert.cgi

Path: //punky

View: Hide editions — Hide workareas

Mount Point:

Locale:

Backing Store:

Target Directory:

Conversion Action: Preview the Conversion Script

Options: Fully clone each edition<br>Clean up history when done

Script Preview:

Status: No conversion is active

//punky
default

| Name | Size | Type | Modified |
|---|---|---|---|
| default | | Directory | Wed Dec 31 16:00:00 1969 |

4. In the **Path** field, enter the network version path (`vpath`) to the TeamSite server containing the source backing store.

   If the source backing store is located on a remote server you must begin the path with two forward slashes (`//`). For example, in the previous graphic, the old-format backing store to be converted is located on a remote server named `punky`.

5. Click **Set** to display the source tree of the `vpath` of the source backing store.

6. Set the **View** menus to filter the files that are displayed in the directory tree view of the source backing store (lower left frame in the conversion GUI).

   The options are:

   - **Hide editions**—Individual editions are not displayed. Converting this branch converts all corresponding editions.
   - **Show editions**—Displays individual editions. You must select which editions are converted. (Expanding an `EDITIONS` directory containing a large number of editions can be very slow.)
   - **Show recent editions**—Displays the 10 most recent editions. You must select which editions are converted.
   - **Hide workareas**—Workareas associated with the selected branch are not displayed or converted.
   - **Show workareas**—Individual workareas are shown. You must select which workareas are converted. (Expanding a `WORKAREAS` directory containing a large number of workareas can be slow.)

7. Click **Set** to update the source tree if you modified the **View** menus (in step 6).

8. In the **Mount Point** field, enter the file system path to the TeamSite virtual file system of the source machine, for example, `F:/`.

9. In the **Locale** field, enter the locale of the backing store to be converted, or leave the field blank to use the default locale.

   This entry is used to specify how non-ASCII metadata is interpreted. The following are valid locales:

   - iso-8859-1
   - iso-8859-2
   - iso-8859-3
   - iso-8859-4
   - iso-8859-5
   - iso-8859-6
   - iso-8859-7
   - iso-8859-8
   - iso-8859-9
   - iso-8859-15
   - euc-jp
   - euc-tw
   - euc-cn
   - euc-kr
   - shift_jis
   - big5
   - gb2312
   - utf-8
   - utf8

10. In the **Backing Store** field, enter the file system path to the source backing store, for example: `//punky/iw-store/default`.

    **Note:** This field is optional, but providing the path results in a significantly faster conversion.

11. In the **Target Directory** field, enter the name of the directory where the new backing store is to be created, for example: `C:/iw-store/newStore1`.

12. Select one of the following actions from the **Conversion Action** menu:

    - **Preview the Conversion Script**—Displays the `iwconvert` command that is generated based on the selected branches and options in the Script Preview field when **Start** is clicked.
    - **Generate the Conversion Script**—Displays the `iwconvert` command that is generated based on the selected branches and options in the Script Preview field, and writes the command to a script file when **Start** is clicked.

      The script files use the `iwconvert_##.sh` naming convention (where # represents an integer) and is created in the `C:/iw-home/local/logs` directory on the target system.
    - **Run Conversion**—Displays the `iwconvert` command that is generated based on the selected branches and options in the Script Preview field, writes the command to a script file, and invokes the `iwconvert` command when **Start** is clicked.

13. Select none, one, or both of the following `iwconvert` options (use Shift+click to select both options):

    - **Fully clone each edition**—Runs `iwconvert` with the `-f` option, so that editions are cloned without any submit event history. Use this option if there are gaps in the set of editions that you are converting, or if you do not want to save the submit history.
    - **Clean up history when done**—Runs `iwconvert` with the `-c` option to generate submit events correctly.

    If you do not select one or both of these options, all history is copied.

14. Click **Start** to initiate the action defined in step 12 and step 13 and to display the status of the conversion:

    - **Status**—Displays whether or not `iwconvert` is currently running from the GUI. If "iwconvertserver not enabled" is displayed, you must run the `iwconvertserver` process that supports the GUI (described in "Administration CLTs" on page 36).
    - **Conversion Step**—Displays the current `iwconvert` command line if it has been started from the GUI.
    - **Conversion Detail**—Displays the progress of the conversion if `iwconvert` has been started from the GUI.

15. If files are submitted during the conversion procedure, you will need to freeze the server (by using `iwfreeze`), create a new edition that contains these files, and repeat the conversion procedure.

    **Note:** The `iwconvert` program creates temporary workareas (`temp_workarea`) in each converted branch as a by-product of the conversion process. You should manually delete these after the conversion.

## Converting Backing Stores from the Command Line

The TeamSite installation program installs a set of command-line tools in the `iw-home\bin` directory. All of these tools are documented in the *TeamSite Command Line Tools* manual that corresponds with your platform. For convenience, the new CLTs are also included in this document. The `iwconvert` CLT is described in this section the other new CLTs are described in “Administration CLTs” on page 36.

### iwconvert.exe Command-Line Tool

The `iwconvert` CLT converts old-format (TeamSite 4.5.x and 5.0.x) backing stores to the new high-performance backing store format.

Complete the following steps to optimize the conversion process.

- Upgrade your source machine to TeamSite 4.5.1 Service Pack 2, or TeamSite 5.0.1 Service Pack 2 (or higher), with all available patches.
- Ensure you have the most recent version of the `iwconvert` tool.

  Updates to the `iwconvert` and `iwmigrate` tools shipped with TeamSite 5.5.1 will be available on the Interwoven support website. Before using either CLT, check the Interwoven support website (`http://support.interwoven.com`) to ensure you have the most recent version of each tool.
- Before running the `iwconvert` command, run `iwfsck -d` on your source backing store to prepare for conversion. The `iwfsck` CLT is described in the *TeamSite Command-Line Tools* manual.

### Options

The following options are valid for the `iwconvert` command:

`-h`

Displays the usage message.

`-v`

Displays the version number.

`-b` *`branch_vpath`*

Location of the branch that contains the editions or workareas to be converted. If the `vpath` begins with *`//hostname/`* the branch is located on a remote TeamSite server.

**Note:** Branches are converted recursively—all editions in subbranches under the specified branch are also converted unless the `-d` option is specified.

`-c`

Cleans up the history information of a previously converted backing store. Requires that `-o` is also specified.

This `iwconvert` step must be performed last, as a separate step, because it may have interbranch dependencies. Note that this action may safely be executed multiple times.

`-d`

Do *not* recursively convert subbranches.

`-s` *`starting_edition`*

Specifies the first edition in a range of editions to be converted for the specified branch (the default is `INITIAL`). Requires that `–b`, `-d`, `-o`, and `–m` are also specified.

`-e` *`ending_edition`*

Specifies the last edition in a range of editions to be converted for the specified branch (the default is the most recent edition). Requires that `–b`, `-d`, `-o`, and `-m` are also specified.

`-f`

Forces a full clone of each edition without the history of submit events for the editions.

Use this option if there are gaps in the set of editions that you are converting, or if you do not want to save the submit history.

`-l locale`

Specifies the native locale of the backing store being converted and how non-ASCII metadata is interpreted. If this option is not specified it defaults to `LC_LOCAL`.

`-m` *iwmnt_mount_point*

Specifies the mount point for the existing (source) `iwserver` installation. Required with `-b`, `-r`, and `-w`.

`-n` *old_backing_location*

Use direct access to the old backing store for faster conversions. Can be used with `-b`, `-r`, or `-w`.

If this option is not specified, `iwconvert` will run slowly due to calls to `sci_GetPredecessors()`.

`-o` *new_backing_location*

Location of the new backing store. This must be a path to the store root. For example, the store named `default` is specified by: `/local/iw-store/default`.

`-r` *workarea_name*

Converts the specified workarea for the branch.

`-w` *workarea_name*

Converts the specified workarea. Requires that `-b`, `-o`, and `-m` are also specified.

`-x`

Increases the verbosity level. Maximum verbosity is level 3, expressed as `-x -x -x`.

`Ctrl+c`

Stops `iwconvert` at the end of the edition currently being converted. When you restart the conversion, `iwconvert` ignores the editions in the branch that have already been converted and converts the remaining editions.

### Usage Summary

- Convert editions:
  ```
  iwconvert -o new_backing_store_location -m iwmnt_mount_point
  [-n old_backing_store_location] -b branch_vpath [-d [-s
  starting_edition] [-e ending_edition]]
  ```
- Convert a workarea:
  ```
  iwconvert -o new_backing_store_location -m iwmnt_mount_point
  [-n old_backing_store_location] -b branch_vpath -w workarea_name
  ```
- Clean up history:
  ```
  iwconvert -o new_backing_store_location -c
  ```

### Example

```
iwconvert -m f:/ -o d:/iw-store/Safari -n C:\iw-store\default
-b //bgunn/default/main/www
```

## Conversion Procedure

Convert each branch (and its associated subbranches) by completing the following procedure.

1. Map a network drive to the source backing store (typically `c:/iw-store/default`) from the target server.
2. Map a network drive to the source server (`\\`*`hostname`*`\iwserver`).
3. Decide which editions are to be converted.

   Branches are converted recursively—all editions in the subbranches under the specified branch are also converted unless the `-d` option is specified. You can convert an individual edition (typically the most recent) or, if the `-d` option is specified, a range of editions.

   For example, the range of `INITIAL` to `ed_0006` would convert seven editions: `INITIAL` and `ed_0001` through `ed_0006`. Each range of editions converted requires a separate invocation of `iwconvert`.
4. If files have been submitted to the staging area since the last edition was published, publish a new edition.
5. Issue the `iwconvert` command from the `iw-home\bin` directory:

   ```
   iwconvert -o new_backing_store_location -m /mount_location -b
   source_branch -d -s start_of_editon_range -e end_of_editon_range
   ```

   For example, using the example edition range from step 3, on a remote server named `factotum`, and a branch named `default/main/intranet`:

   ```
   iwconvert -o c:\iw-store\default -m F:\ -b
   \\factotum\default\main\intranet -d -s INITIAL -e ed_0006
   ```

   **Note:** You should save the `stdout` and `stderr` output from the `iwconvert` procedure to a log file by appending the following to the command in this step:
   `...INITIAL -e ed_0006`
6. Convert the changes that occurred while the conversion was running by performing either of these steps:
   - Submit all changes from workareas to staging, then publish a new edition that contains these changes. Convert this new edition as described in step 5.

- Convert again using the `-w` option to convert the workareas that contain changes not submitted to the staging area before the conversion described in step 5.

  ```
  iwconvert -o c:\iw-store\default -m F:\ -b
  \\factotum\default\main\intranet -w jerome
  ```

  If you have changes in a large number of workareas it is easier to have users submit their changes and publish and convert a new edition rather than converting the workareas that contain changes.

  This step should complete much faster than your original conversion.

7. Freeze your source server by running the `iwfreeze` command from the `iw-home/bin` directory specifying a large number of seconds for the freeze, for example:

   ```
   iwfreeze +50,000
   ```

8. Repeat step 6 to convert the final changes that were made during the second conversion.
9. Run `iwconvert` with the `-c` option to clean up the second-predecessor links in the new-format (target) backing store, for example:

   ```
   iwconvert -o c:\iw-store\default -c
   ```

   These links are created by TeamSite operations including `Copy To`. The clean up of history must be done as a separate pass at the end of other `iwconvert` passes because the second-predecessor links can point in various directions between branches and areas in the backing store, and the referenced objects may not be converted at the time they are needed.

   If the workareas contain versions of files that have not been converted, the correct contents of those files are copied into the workarea, and those files will show up as modified.

Note the following:

- The `iwconvert` program creates temporary workareas (`temp_workarea`) in each converted branch as a by-product of the conversion process. You should manually delete these after the conversion.

- Do not use the Registry Editor to point to the newly converted backing store. While you can move your default store by editing the registry, this method is not reliable and only works when the directory is named `default`. If you use this method and convert to a destination directory with a subdirectory named something other than `default`, the new store will not be activated.

## Creating Multiple Backing Stores

Multiple backing stores can be created using two different methods depending on where you want to locate them, and whether you want to use multibyte characters in their names.

- `iwstoreadm` CLT—Creates and activates new backing stores when issued with the `-a` option.
  - Accepts ASCII characters for store names.
  - Creates the new backing store in the default location (typically `C:\iw-store\`).
  - Does *not* allow for a descriptive comment to be added to the backing store.
- Editing the `iw.cfg` file—Defines backing stores with entries in the `[iwserver]` section of the `iw.cfg` file.
  - Accepts multibyte characters for the store name (though the path to the store must use ASCII characters)
  - Creates the new backing store in any location.
  - Allows you to add a descriptive comment to the backing store. This comment is displayed when the active backing stores are listed from the command line, or displayed in the TeamSite GUIs.
  - Must be activated by using the `iwstoreadm` CLT with the `-a` option.

  If you want to define backing stores by editing the `iw.cfg` file, complete the procedure described in the next section. If you want to create backing stores using the `iwstoreadm` CLT, complete the procedure described in "Creating Backing Stores Using the iwstoreadm CLT" on page 35.

## Defining Backing Stores in the iw.cfg File

As previously mentioned, the advantages of defining backing stores in the `iw.cfg` file include the ability to use multibyte characters in store names and to locate the backing store in a directory other than `C:\iw-store\`.

User-defined backing stores which are named using multibyte characters, must have a corresponding entry in the `iw.cfg` file. While the name of the backing store can be defined in multibyte characters, the backing store location *must* be defined using ASCII characters. All backing store data is stored in UTF-8 encoding.

Complete the following procedure to create backing stores defined in the `iw.cfg` file:

1. Ensure that the user you are logged in as has an entry in the Master role file (*`iw-home`*`\conf\roles\master.uid`).
2. Open the `iw.cfg` file in a text editor.

   By default, the `iw.cfg` file is located in *`iw-home`*`\etc`.
3. If you are using multibyte characters for the store name, specify the encoding of your `iw.cfg` file by creating the following entry as the first line in the file—it *must* be the first line or it will be ignored.

   ```
   [iwcfg]
   encoding=locale_name
   ```

   where *`locale_name`* is one of the following locales:

   - shift-jis (Japanese)
   - cp1252 (French or German)

   For example:

   ```
   [iwcfg]
   encoding=shift-jis
   ```

   **Note:** The locale entry must match the encoding of your text editor. Refer to page 342 for details about text editor encodings.
4. Append the following entry to the `[iwserver]` section to define additional backing stores:

   ```
   store_directory_store_name=absolute_path_to_backing_store
   ```

For example:

```
store_directory_salesAsia=C:\salesAsia
```

**Note:** The *absolute_path_to_backing_store* must be in ASCII while the *store_name* and the optional *descriptive_comment* (described in step 5) can be in high-ASCII or multibyte characters.

5. Optionally, add a comment to the `[iwserver]` section below the backing store you just defined:

   ```
   store_comment_store_name=descriptive_comment
   ```

   For example:

   ```
   store_comment_salesAsia=Store for Demo
   ```

   The completed entry, should look like this:

   ```
   [iwserver]
   existing iwserver entries
   store_directory_salesAsia=C:\salesAsia
   store_comment_salesAsia=Store for Demo
   ```

6. Save and close the `iw.cfg` file.

7. Run the `iwreset` CLT to have the TeamSite server read the changes to the `iw.cfg` file.

8. Run the `iwstoreadm` CLT with the `-a` option to create the newly defined backing store:

   ```
   >iwstoreadm -a salesAsia
   ```

   The `iwstoreadm` CLT checks the `iw.cfg` file to see if a `store_directory` or `store_comment` entry exists, when it finds these entries, their definitions are used to create the backing store.

   The system then activates and mounts the new backing store.

9. Run the `iwstoreadm` CLT with the `-l` option to list all active backing stores:

   ```
   >iwstoreadm -l
   ```

The system displays the following:

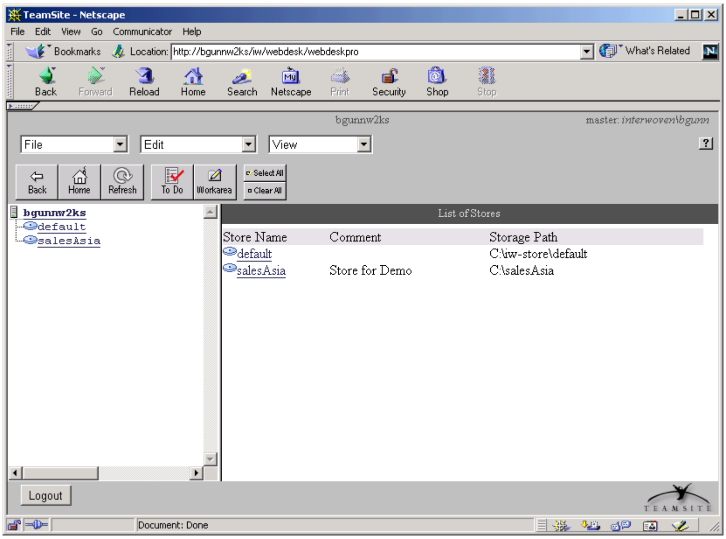
```
Name        Store Directory              ID          Comment
----        ---------------              --          -------

default     C:\local\iw-store\default    0x64
salesAsia   C:\salesAsia                 0x65        Store for Demo
```

10. Open your web browser and log in to WebDesk Pro or WebDesk.

11. Click **Workarea** (WebDesk Pro) or the **Files** tab (WebDesk).

    The backing store and comment you created is listed in the GUI.


TeamSite - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://bgunnw2ks/iw/webdesk/webdeskpro
What's Related
Back Forward Reload Home Search Netscape Print Security Shop Stop
bgunnw2ks
master: interwoven\bgunn
File
Edit
View
?
Back Home Refresh To Do Workarea
Select All
Clear All
bgunnw2ks
default
salesAsia
Logout
TEAMSITE
Document: Done

List of Stores

| Store Name | Comment | Storage Path |
|---|---|---|
| default | | C:\iw-store\default |
| salesAsia | Store for Demo | C:\salesAsia |

**Notes:**

- You can repeat the procedure to create any number of backing stores, but you can only have eight active at one time.
- You can edit the `store_directory_`*`storename`* entries to move backing stores defined in `iw.cfg`.

## Creating Backing Stores Using the iwstoreadm CLT

The following procedure describes the creation of backing stores from the command line using `iwstoreadm`. It also describes viewing the newly created backing stores in both the command window and the TeamSite WebDesk Pro interface.

1. Ensure that the user you are logged in as has an entry in the Master role file (*iw-home*`\conf\roles\master.uid`).

2. Issue the `iwstoreadm -a` *store_name* command to create a new store, for example:

   ```
   >iwstoreadm -a store1
   ```

   `store1` is created in `C:\iw-store\` and activated.

3. Type the following command to list the active backing stores:

   ```
   >iwstoreadm -l
   ```

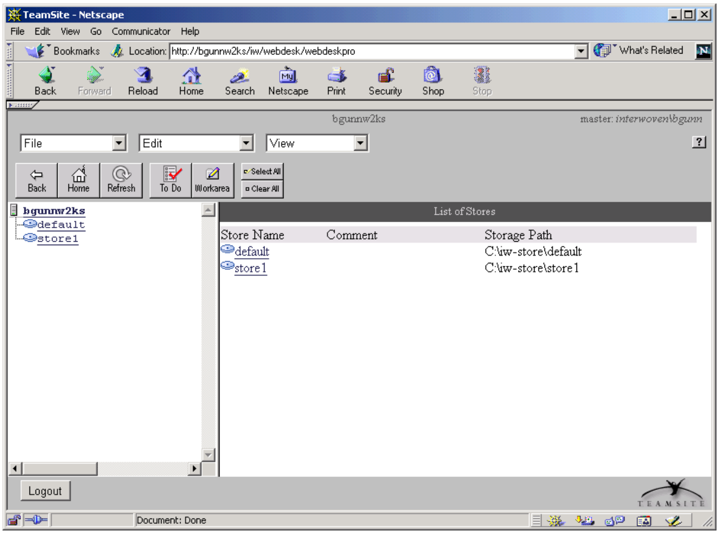
   The following listing is displayed:

   ```
   Name        Store Directory        ID        Comment
   ----        ---------------        --        -------
   default     C:\iw-store\default    0x64
   store1      C:\iw-store\store1     0x65
   ```

4. Open your web browser and log in to WebDesk Pro or WebDesk.

5. Click **Workarea** (WebDesk Pro) or the **Files** tab (WebDesk).

The backing store you created is listed in the GUI.


TeamSite - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://bgunnw2ks/iw/webdesk/webdeskpro
What's Related
Back Forward Reload Home Search Netscape Print Security Shop Stop
bgunnw2ks
master: interwoven\bgunn
File
Edit
View
Back Home Refresh To Do Workarea
Select All
Clear All
bgunnw2ks
default
store1
Logout
TEAMSITE
Document: Done

List of Stores

| Store Name | Comment | Storage Path |
|---|---|---|
| default | | C:\iw-store\default |
| store1 | | C:\iw-store\store1 |

6. Click `store1`.

   Note that all backing stores (including the system-generated `default`) contain a `Main` branch, and a `STAGING` area that is based on an `INITIAL` edition.

# Administration CLTs

This section describes the new command-line administration utilities designed for use with the conversion GUI and new backing store functionality. The conversion CLT (`iwconvert`) is described on page 26.

## iwstoreadm.exe

Backing store administration involves creating, activating, and deactivating backing stores by using the `iwstoreadm` CLT. When the `iwstoreadm -a` *`storename`* command is issued, the following sequence is triggered:

- The `iw.cfg` file is checked to see if a `store_directory` or `store_comment` entry exists, if it does, their definitions are used to create the backing store. If these entries do not exist:

- The backing store directory is automatically created in `C:\iw-store\` and populated with a minimal backing store containing a branch named `main`. The store remains active until explicitly deactivated by using `iwstoreadm -d` (even if the server is stopped and restarted).

  Deactivating a store does *not* delete it. A store can be deactivated, moved to a new location, and reactivated using original name though it will be assigned a new store ID.

### iwstoreadm.exe Options

The following options are valid for the `iwstoreadm` command:

-a *store_name* — Activates an existing store, or creates and activates a new store. You must have an entry in the Master role file (*iw-home*\conf\roles\master.uid) to execute this option.

-d store_*name* — Deactivates an existing store. You must have an entry in the Master role file (*iw-home*\conf\roles\master.uid) to execute this option.

-h — Displays the usage message.

-l — Lists active stores.

### Usage

iwstoreadm [-l] [-a *store_name*] [-d *store_name*]

### Example

>**iwstoreadm -l**

Displays the active backing stores:

```
Name      Store Directory        ID         Comment
----      ---------------        --         -------
default   C:\iw-store\default    0x64
store2    C:\iw-store\store2     0x65
```

## iwidmap.exe

A command line tool called `iwidmap` is included to change the mapping between the SID and the token. It can also be used to refresh the mapping when the same names are used, and the SID has changed. For more information about SID mapping, refer to the *TeamSite Administration Guide*.

### Usage

```
iwidmap [-v] [-h] (-u | -g) [-a][-c <user1> <user2>] [-x <file> | -i <file>]
backing-store
```

| | |
|---|---|
| `-v` | Displays the version of this program. |
| `-h` | Displays the usage message. |
| `-u` | Update `userid` mapping. |
| `-g` | Update `groupid` mapping. |
| `-a` | Update all entries in the ID map. |
| `-c <user> <user2>` | Update user1 to user2. |
| `-x <file>` | Extract to file. |
| `-i <file>` | Import from file. |
| `backing-store` | Location of the backing store. |

### Example

```
iwidmap -u -c jgarcia rhunter c:\iw-store\NewReleases
```

## iwmigrate.exe

The `iwmigrate` CLT is similar to `iwconvert` except that it accepts new-format backing stores as its source. It can be used to split a single new-format backing store into multiple backing stores, or to move the contents of a store to another location without losing the history of submit events for the editions.

**Note:** Updates to the `iwconvert` and `iwmigrate` tools shipped with TeamSite 5.5.1 will be available on the Interwoven support website. Before using either CLT, check the Interwoven support website (`http://support.interwoven.com`) to ensure you have the most recent version of each tool.

### Usage

```
iwmigrate [-h] [-v] [-x] [-m mount_location] -o new_backing_location
[-b branch_vpath] [-s starting_ed] [-e ending_ed] [-n old_backing_location]
[-w workarea_name] [-c] [-d] [-f] [-l]
```

`-h` Display this message.

`-v` Display version number.

`-b branch` Specify source branch for migration.

`-x` Increase verbosity level. Maximum verbosity is level 3, expressed as `-x -x -x`.

`-m mount_location` Specify mount location of backing store, for example: `F:\`

`-o new_backing_location` Specify new backing store location.

`-n old_backing_location` Specify old backing store location.

`-d` Do *not* recursively convert subbranches

`-s starting_ed` Specify starting edition for migration. Default is the `INITIAL` edition (this option can only be used with `-d`)

`-e ending_ed` Specify ending edition for migration. Default is the most recent edition.

`-f` Full clone of every edition (does *not* preserve history).

`-r` Clean up history information (use on the final pass).

`-l locale` Specify the native locale of the backing store being migrated (if different from `LC_LOCAL` for this system).

### Example

```
iwmigrate -m F:\Safari -o d:\iw-store\safari_on_line -b
```

## iwconvertserver.exe

The `iwconvertserver` process supports the conversion GUI by communicating with remote TeamSite servers and invoking `iwconvert` on behalf of the GUI. You must run the `iwconvertserver` process manually for the GUI to function properly.

1. Change to the `iw-home/bin` directory, for example:

   ```
   >cd C:\Program Files\Interwoven\TeamSite\bin
   ```

2. Either:

   - Run the `iwconvertserver` utility directly from the command line:
     **`iwconvertserver -S`**
   - Install it as a service, and then start the service using the Service Control Manager:
     **`iwconvertserver -is`**